APPENDIX

1-2. (canceled)

3. A processor, comprising:

a register set;

at least one execution unit that executes load instructions to transfer data into said register set;

a load queue containing at least one entry, wherein said entry stores load data retrieved by a first load instruction; and

queue management logic that, responsive to execution of a second load instruction, detects by reference to said load queue whether a data hazard exists, and if so, outputs said load data retrieved by said first load instruction from said entry to said register set in accordance with said second load instruction.

4. The processor of Claim 3, wherein said entry stores a target address of said first load instruction and has a hazard flag indicative of a possible data hazard, wherein said queue management logic detects that a data hazard exists if said second load instruction precedes said first instruction in program order and a target address of said second load instruction matches said target address stored in said entry and said hazard flag is set.

5. The processor of Claim 4, wherein said queue management logic sets said hazard flag at least in response to local store operation specifying said target address.

6. The processor of Claim 3, said register set comprising a general purpose register set.

7. The processor of Claim 3, wherein said queue management logic outputs said load data from said load queue to a register in said register set that is specified by said second load instruction.

8. The processor of Claim 3, wherein said queue management logic, responsive to detection of a data hazard, initiates reexecution of at least said first load instruction but not said second load instruction.

9. The processor of Claim 3, wherein said queue management logic allocates a respective entry within said load queue to each load instruction upon dispatch and, upon completion of said each load instruction, deallocates said respective entry.

10. A data processing system, comprising:

an interconnect fabric;

a memory coupled to said interconnect fabric;

a register set;

at least one execution unit that executes load instructions to transfer data from said memory into said register set;

a load queue containing at least one entry, wherein said entry stores load data retrieved by a first load instruction; and

queue management logic that, responsive to execution of a second load instruction, detects by reference to said load queue whether a data hazard exists, and if so, outputs said load data retrieved by said first load instruction from said entry to said register set in accordance with said second load instruction.

11. The data processing system of Claim 10, wherein said entry stores a target address of said first load instruction and has a hazard flag indicative of a possible data hazard, wherein said

queue management logic detects that a data hazard exists if said second load instruction precedes said first instruction in program order and a target address of said second load instruction matches said target address stored in said entry and said hazard flag is set.

12. The data processing system of Claim 11, wherein said queue management logic sets said hazard flag at least in response to local store operation specifying said target address, wherein said local store instruction is earlier in program order but later executed than said first load instruction.

13. The data processing system of Claim 12, wherein said at least one execution unit, said register set and said load queue comprise a first processor and said data processing system includes a second processor, wherein said queue management logic also sets said hazard flag in response to said second processor issuing an exclusive access operation specifying said target address on said interconnect fabric.

14. The data processing system of Claim 10, said register set comprising a general purpose register set.

15. The data processing system of Claim 10, wherein said queue management logic outputs said load data from said load queue to a register in said register set that is specified by said second load instruction.

16. The data processing system of Claim 10, wherein said queue management logic, responsive to detection of a data hazard, initiates reexecution of at least said first load instruction but not said second load instruction.

17. The data processing system of Claim 10, wherein said queue management logic allocates a respective entry within said load queue to each load instruction upon dispatch and, upon completion of said each load instruction, deallocates said respective entry.

18. A method of executing load instructions out-of-order in a processor having a register set and a load queue, said method comprising:

storing, in an entry in said load queue, load data retrieved from memory in response to executing a first load instruction;

in response to execution of a second load instruction, detecting by reference to said load queue whether a data hazard exists; and

in response to detection of a data hazard, outputting said load data retrieved by said first load instruction from said entry to said register set in accordance with said second load instruction.

19. The method of Claim 18, wherein said entry stores a target address of said first load instruction and has a hazard flag indicative of a possible data hazard, wherein detecting that a data hazard exists comprises determining if said second load instruction precedes said first instruction in program order and a target address of said second load instruction matches said target address stored in said entry and said hazard flag is set.

20. The method of Claim 19, and further comprising setting said hazard flag at least in response to a local store operation specifying said target address, wherein said local store instruction is earlier in program order but later executed than said first load instruction.

21. The method of Claim 19, wherein outputting said load data comprises outputting said load data from the load queue to a register in said register set that is specified by said second load instruction.

22. The method of Claim 19, and further comprising:

in response to detection of a data hazard, initiating reexecution of at least said first load instruction but not said second load instruction.

23. The method of Claim 19, and further comprising allocating a respective entry within said load queue to each load instruction upon dispatch and, upon completion of said each load instruction, deallocating said respective entry.